Product family optimization: a multiplatform algorithm based on iterative increase of the commonality

Derrick Fongang Fongang, Rodolphe Le Riche, Xavier Bay

Product family optimization: a multiplatform algorithm based on iterative increase of the commonality

Derrick Fongang Fongang[1,2], Rodolphe Le Riche[1,2], Xavier Bay[1,2]

[1] Inst. H. Fayol , Ecole des Mines de Saint-Etienne
158, Cours Fauriel, 42 023 Saint-Etienne cedex, France
{fongang, leriche, bay}@emse.fr
[2] CNRS UMR 6158 LIMOS

Keywords : *product family, product platform, multicrieria optimization, mixed-integer non linear programming.*

1 Context

This work addresses the problem of optimizing the design of a family of products and simultaneously maximizing the commonality between these products. By commonality, we mean the proportion of components that are shared between the products. The set of all components common to all products is called platform [2, 3]. We consider here the multiplatform problem that allows the commonality to start as soon as one component is common to two products. The multiplatform formulation encompasses every possible case of commonality and leads to a highly complex optimization problem: for N_P products, of each N design variables, there are $(B_{N_P})^N$ multiplatform configurations (B_{N_P} is the N_P^{th} number of Bell). In this paper, we propose an algorithm to tackle this multiplatform product family optimization. Similarly to [1, 3] , we will estimate the Pareto front between an aggregation of the products engineering performance and the family commonality. The proposed method is based on a quadratic approximation of the product performance functions and has a complexity of $O(N \times N_P^2 \times \log(N_P))$, which is much better than the complexity of any enumeration strategy $((B_{N_P})^N)$.

2 Mathematical formulation

Considering a family of N_P products, we call $x^i \in \mathbb{R}^N$ and f^i, $i = 1, \ldots, N_P$, the design attribute values and the performance functions of the products P^i respectively. Let V be a list of N matrices $v_j = (v_j^{i_1,i_2})$, $(1 \leq j \leq N$ and $1 \leq i_1, i_2 \leq N_P)$, that described the commonality within the products family.

$$v_j^{i_1,i_2} = \begin{cases} 1, & \text{if } x_j^{i_1} = x_j^{i_2} \\ 0, & \text{otherwise} \end{cases} \quad \text{with } 1 \leq i_1, i_2 \leq N_P \tag{1}$$

For example, in the automotive industry, P^i could be a manufacturer's car range, x^i the car body parameters of P^i, and f^i its crash-worthiness. Maximizing the family commonality is equivalent to minimize the number of degrees of freedom D, that is equal to the number of independant variables. The Equation 2 hereafter gives a mathematical formulation of this problem.

$$\begin{cases} & \min\limits_{X,V} \quad D(V) \\ \text{and} & \min\limits_{X,V} \quad \sum\limits_{i=1}^{N_P} f^i(x^i) \\ & \text{where} \quad X = (x^1, x^2, \ldots, x^{N_P}) \\ & \text{and} \quad V = (v_1, v_2, \ldots, v_N) \\ & \text{s.t.} \quad v_j^{i_1,i_2}.(x_j^{i_1} - x_j^{i_2}) = 0 \end{cases} \tag{2}$$

A two-stage enumeration algorithm consists in first sampling all the possible partitions (all the possible V values), second solving for each V the continous optimization problem in X and keeping the optimal solutions as points of the Pareto front. In Figure (1), each point represents one value of (V, X), with its corresponding number of degrees of freedom D on the x-axis, and its aggregated performance function on the y-axis.

3 One step ReDuction Algorithm (ORDA) in one dimension

To reduce the number of degrees of freedom by one, we should first choose one design variable, x_j, and then select two products P^{i_1} and P^{i_2} that will share the same value for that design variable, which means $v_j^{i_1,i_2} = 1$ and $x_j^{i_1} - x_j^{i_2} = 0$. In this section, we suppose the parameter x_j already chosen (so that we temporarily drop the subscript). The idea behind the ORDA algorithm is to order all the possible pairs of products, and to pick the pair (k, l) that increases the least the aggregated engineering performance function when merged. Let q^i be a quadratic approximation of the function f^i around its non-platform optima. The implementation of

Algorithm 1 : ORDA algorithm

Require: Q, the list of all the functions q^i, $1 \leq i \leq N_P$
PQ, the list of $\frac{\#Q(\#Q-1)}{2}$ pairs of functions ordered by increasing $\Delta^{k \bullet l}$(cf. Equations 6 & 3)
Begin
pick the first pair (q^k, q^l) from PQ
create the function $q^{k \bullet l}(x) = q^k(x) + q^l(x)$ by merging the hyper-variables x^k, x^l
delete q^k and q^l from Q
Update PQ:
delete from PQ all the pairs that contain q^k or q^l
insert in the ordered PQ all new pairs obtained with $q^{k \bullet l}$ and other functions from Q
Update Q:
add the function $q^{k \bullet l}$ to Q.
return Q,PQ
End

ORDA is based on the quadratic nature of the functions q^i's:

$$q^i(x^i) = \frac{1}{2} h_i (x^i)^2 + a_i x^i + b_i \tag{3}$$

where $h_i > 0$, a_i and b_i are scalars. When x^k and x^l are common, the new function $q^{k \bullet l}$ is :

$$q^{k \bullet l}(x^{k \bullet l}) = \frac{1}{2}(h_k + h_l)(x^{k \bullet l})^2 + (a_k + a_l) x^{k \bullet l} + (b_k + b_l) \tag{4}$$

Thus, in the ORDA algorithm, to find the pair of variables to set equal, we should solve:

$$\min_{1 \leq k < l \leq N_p} \min_{X} \left(q^{k \bullet l} + \sum_{1 \leq i \neq k,l \leq N_p} q^i \right) \tag{5}$$

where X accounts for the merged k and l variables. Using the Equation 3 and 4, Equation 5 is equivalent to minimizing on k and l, $\Delta^{k\bullet l}$. We denote:

$$\Delta^* = \min_{1\leq k<l\leq N_p} \left(\Delta^{k\bullet l} = -\frac{(a_k + a_l)^2}{(h_k + h_l)} + (\frac{a_k^2}{h_k} + \frac{a_l^2}{h_l}) \right) \tag{6}$$

4 The general ORDA algorithm

Repeated applications of the one dimensional ORDA reduction algorithm can serve as a building block to approximate the commonality-engineering performance Pareto front in the general case. The algorithm starts from the extremity where all the design attributes are different (no platform) and moves step by step by either creating a new two elements platform or by adding one element to an existing platform. These steps can be suboptimal as it is possible that the optimal reduction of one degree of freedom involves a complete reorganization of the platforms.

Generalization of the ORDA algorithm to many dimensions is also straightforwardï¿½: again, the idea is to make a quadratic approximation q^i of the performance function for each product P^i around its no platform optima. We denote by q^i_j the restriction of q^i to its variable x^i_j. The coefficients of the quadratic functions q^i_j which depend on the x^i_k, $k \neq j$, need to be updated. The algorithm 2 summarizes the procedure to follow.

Algorithm 2 : Multi-dimensional repeated ORDA algorithm (rORDA)

Require: Q, the list of all the functions q^i_j, $1 \leq i \leq N_P$, $1 \leq j \leq N$
  PQ, an ordered list of all the pairs of q^i_j functions $1 \leq i \leq N_P$, $1 \leq j \leq N$
  while the targeted commonality is not reached ($D(V) > D^{target}$) **do**
    for all dimensions $t = 1, \ldots, N$ **do**
      apply ORDA to the t^{th} dimension, get the performance degradation Δ^*_t
    end for
    choose the dimension j that optimizes the most, when reducing by one the number of degrees of freedom ($j = arg\min_{t=1,\ldots,N} \Delta^*_t$) (cf. Equation 6)
    Record optimized variables (V^*, X^*)
    update Q and PQ
  end while
  return the (sum of q^i performance, D) Pareto front approximation from the list of recorded (V^*, X^*) variables

Multiplatform products family optimization is a complex problem: with N_P products of N design variables each, there are $(B_{N_P})^N$ possible multiplatform configurations for which a continuous optimization needs to be done. B_{N_P}, the Bell number, is the number of partitions of a set of size N_P and it has a complexity of order $O\left((N_P \log(N_P))^{N_P}\right)$. With $N_P = 11$ and $N = 1$, there are $B_{11} = 678570$ possible partitions (V values). It can be seen that the complexity of the ORDA algorithm is induced by sorting the pairs of merged performances ($\Delta^{k,l}$) and is $O(N_P^2 \log(N_P))$. The overall algorithm complexity is $O(N \times N_P^2 \log(N_P))$. The Figure (1) shows an example of Pareto front and its approximation built with $N_P = 11$ products. The Table 1 describes the associated 11 quadratic functions. The dashed line represents the true Pareto front as obtained by enumerating all possible partitions (i.e., multiplatforms) while the solid line represents the ORDA approximation of the Pareto front. The three first variables that are merged by the ORDA algorithm are optimal, in the sense that the algorithm remains on the true Pareto front. The fourth merged variable, when going from $D = 8$ to 7 degrees of freedom, is suboptimal, yet the approximation errors to the true Pareto front seems bounded as a near optimal solution is found at $D = 5$.

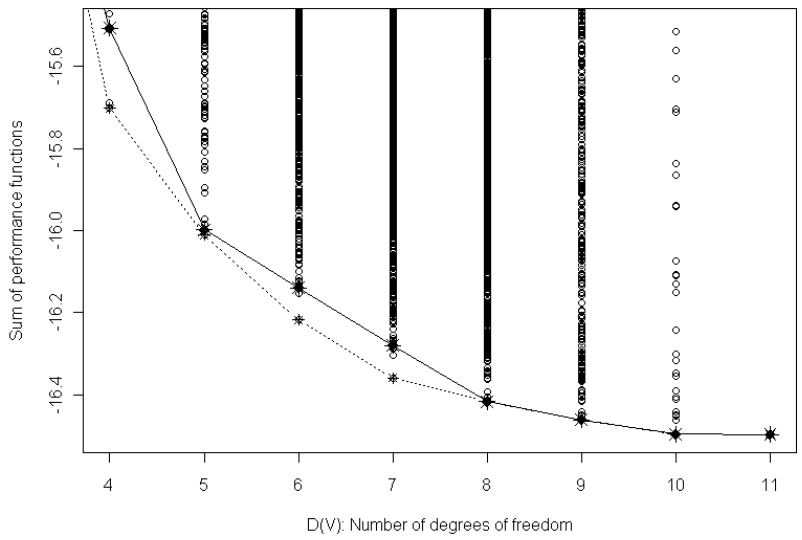

FIG. 1: Example of commonality / performance compromise with 11 quadratic functions. Dashed lineï¿½: the true Pareto front. Solid lineï¿½: approximation of the Pareto front obtained by successive ORDA calls. The functions are described in Table 1.

	q^1	q^2	q^3	q^4	q^5	q^6	q^7	q^8	q^9	q^{10}	q^{11}
h	55	45	75	25	15	85	5	65	12	33	70
a	4.1	8.3	-2.4	9.8	-3.5	-7.2	3.0	1.2	-25.1	15.6	27
b	-3.0	2.0	4.0	3.0	15.0	24.0	-19.0	-15.0	0.15	26.0	-14.0

TAB. 1: Coefficients of the eleven quadratic functions test case.

5 Conclusion and perspectives

We have presented an algorithm named rORDA for optimizing product families in a multi-platform context. The platform benefit was measured in terms of the number of independent variables and needs to be balanced against performance loss due to the platforms. The resulting formulation is a multicriteria mixed-integer nonlinear programming problem. rORDA is a heuristic strategy that approximates the Pareto front of this complex problem by sequentially merging variables on the basis of a local quadratic approximation of the product performances. The numerical complexity is much lower than any enumeration of all platforms and our first tests show good approximations of the Pareto front. The solutions provided by rORDA could typically serve as starting points for other optimization algorithms that do not rely on a quadratic approximation.

References

[1] Souma Chowdhury, Achille Messac, and Ritesh A Khire. Comprehensive product platform planning (CP3) framework. *Journal of Mechanical Design*, 133(10):101004, 2011.

[2] Marc H. Meyer and Alvin P. Lehnerd. *The Power of Product Platforms*. Free Press, November 2011.

[3] T.W. Simpson, Z. Siddique, and J. Jiao. *Product platform and product family design: methods and applications*. Springer Verlag, 2006.

[4] M. Tawarmalani and Nikolaos V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Springer, October 2002.